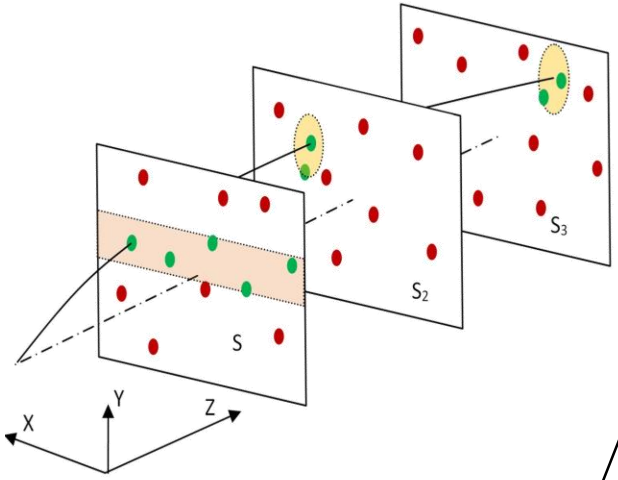
# Track and vertex reconstruction using ML




# Tracking and vertexing

The classical HEP algorithms for track and vertex reconstruction are based on the Kalman Filter (**KF**) method, since KF makes it easy to take into account the inhomogeneity of the magnetic field, multiple scattering and energy losses.

## Event reconstruction

*Event reconstruction plays a key role in data processing for High Energy Physics (HEP) experiments. It consists of two stages: recognizing the tracks and then finding the vertices.*

KF is used sequentially station by station for track recognizing and fitting and then also sequentially for vertex finding.

However, in order to start KF needs a very time consuming preliminary search of the initial set of parameters (so called seeding).

Besides, KF suffers from exponentially growing computational complexity and lack scalability while increasing the event multiplicity

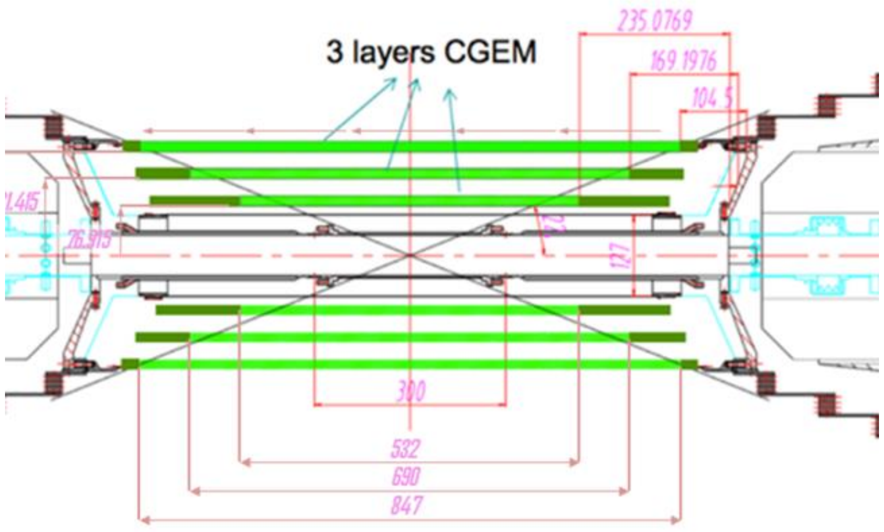
Despite the KF success and many tricks to reduce the seeding time, this method still has several disadvantages caused just by its locality, when tracks are reconstructed one by one. Local approaches have an obvious drawback: they do not allow access to the global picture of an event and see the dependence between individual tracks or groups of tracks.

Also, there is no direct possibility to look for such phenomena as secondary vertices.

**At the same time, there is another global approach, in which the recognition of the entire event including all tracks and vertex itself among noises is performed immediately across the whole picture of this event.**

# Features of detector of the BES-III experiment and our goals

BES-III detector has only three cylindrical GEM stations. This means, if a particle is registered only at two out of three stations, then its track in magnetic field cannot be restored without additional information.

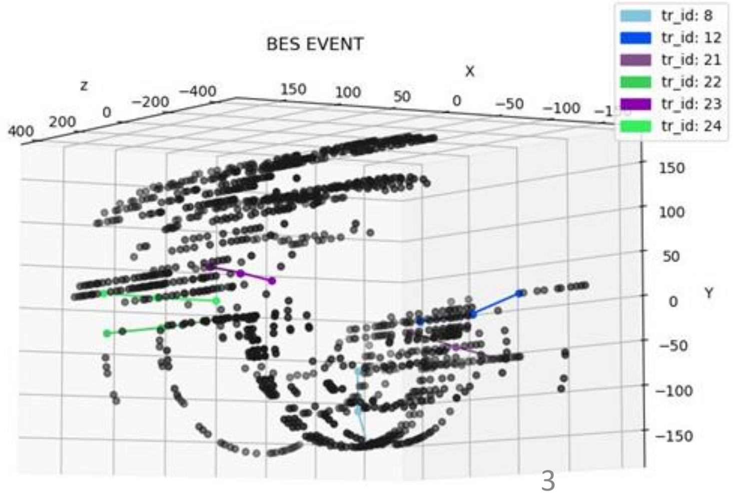



**Knowledge about primary vertex**

Helps to improve the precision in determining the particle momentum.

Leads to the significant reduction of the algorithmic complexity during the track-candidate search – from O($n^2$) to O(n), and can improve the overall track reconstruction efficiency.

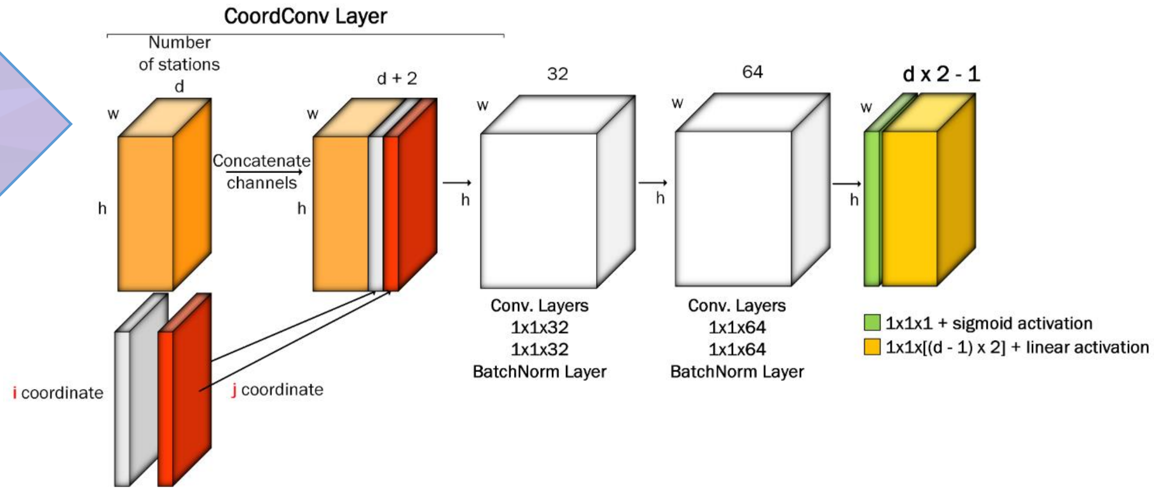
Should also be noted that tracking is especially complicated for strip GEM detectors due to their design specifics, which leads to the appearance of two orders of magnitude more fake hits in addition to useful ones.



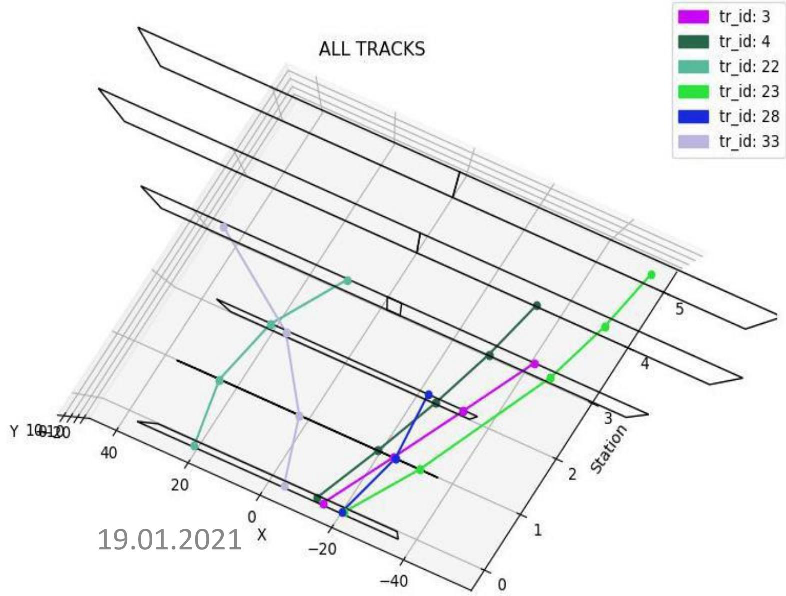
# How to extend a convolutional neural network to represent a physical event

## Using Look Once On Tracks (LOOT) model

Our main idea is to use OZ dimension instead of RGB channels – it's a radically new approach. Height and Width are the sizes of the largest station (most often the last).




See Goncharov et al http://ceur-ws.org/Vol-2507/130-134-paper-22.pdf




## Event - image. Stations - color

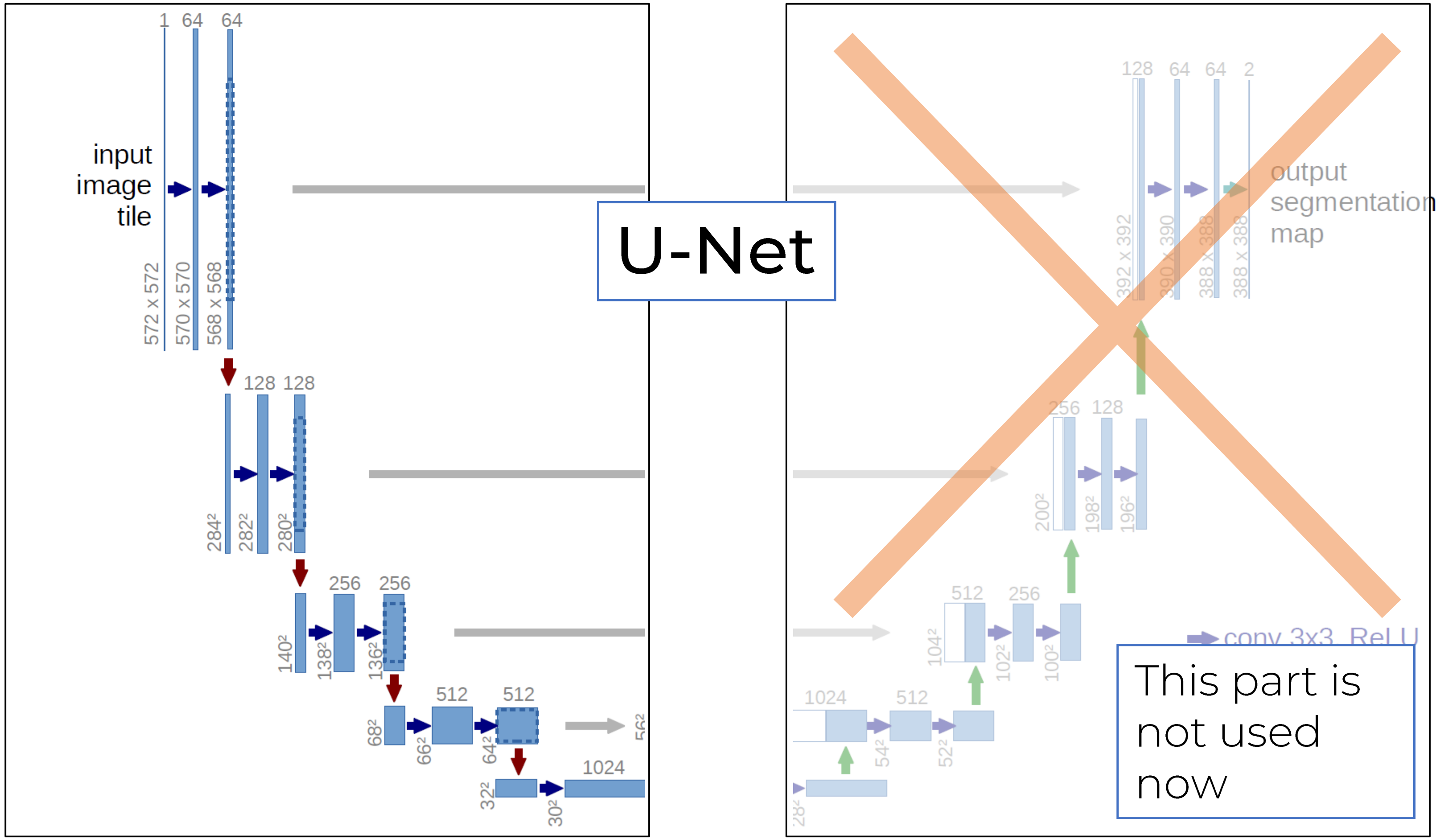
Images have 3D format: Height+Width+RGB

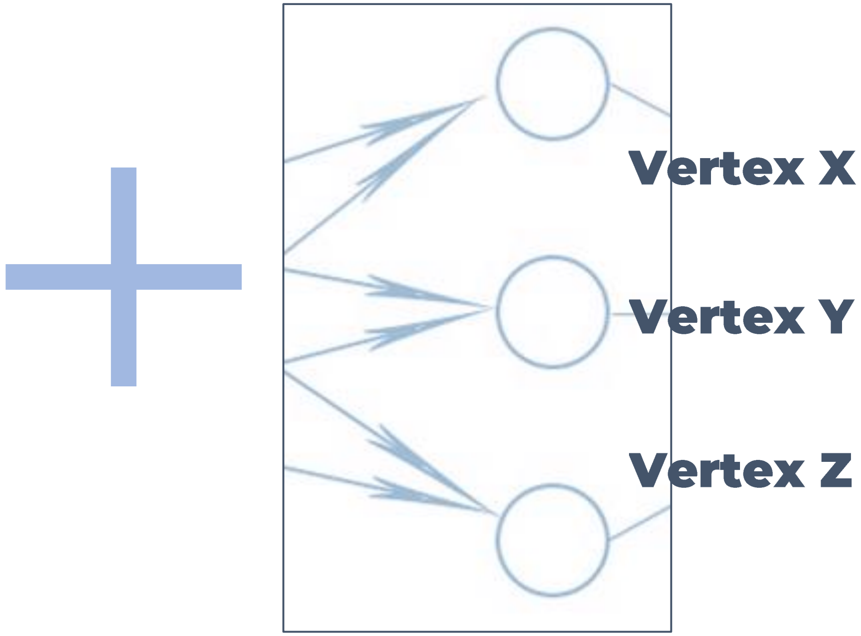
- ✓ Data from each station is a sparse matrix of zeros and ones, where ones indicate hits appearance
- ✓ Events have 3D format too: Height+Width+Stations

# LOOT + U-net architecture for vertex prediction

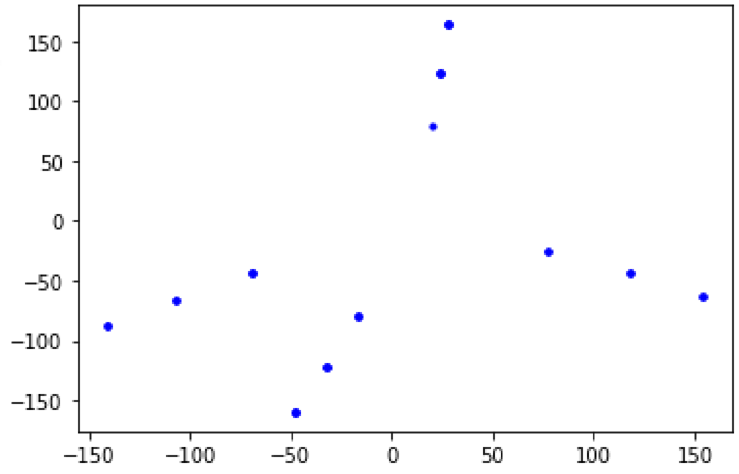
U-Net is a convolutional neural network that was developed for biomedical image segmentation.

Network consists of a contracting path and an expansive path, which gives it the u-shaped architecture.



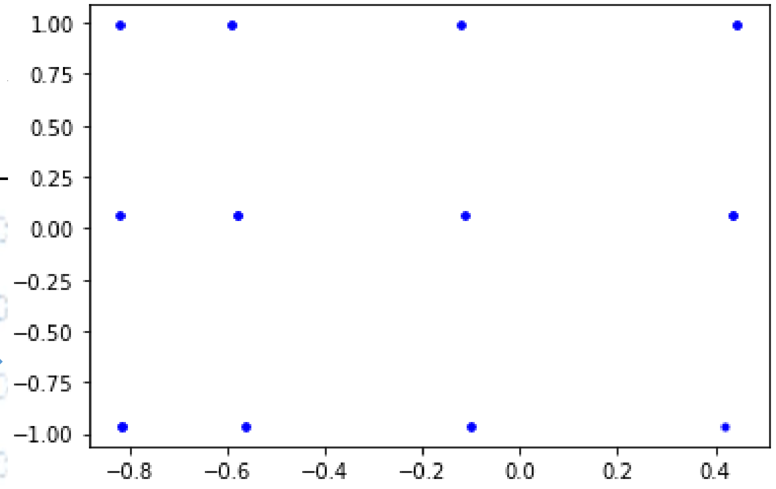
# Data preprocessing



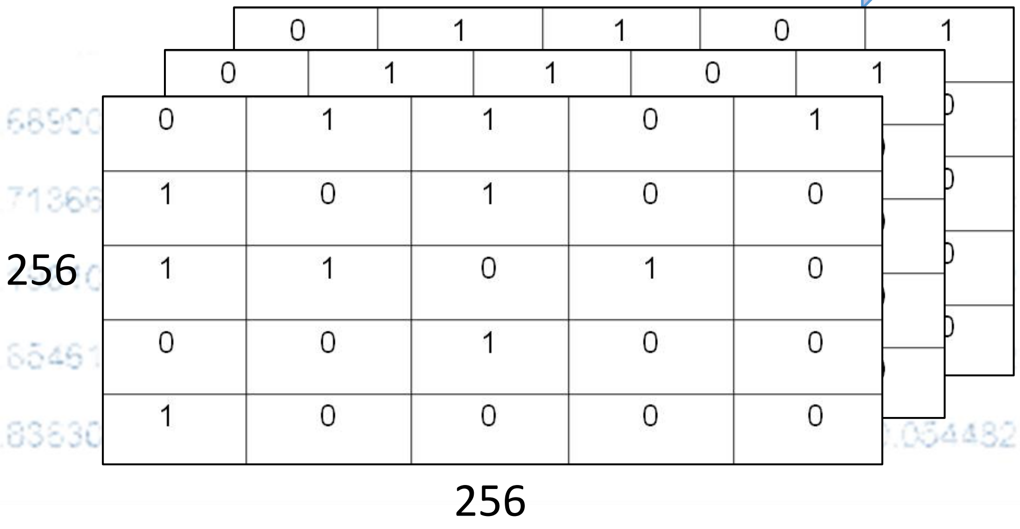

Translation from rectangular to cylindrical coordinates

$$Z = \sqrt{x^2 + y^2},$$
$$X = arctg\left(\frac{y}{x}\right)$$
$$\mathrm{Y} = \mathrm{z}$$

Data normalization

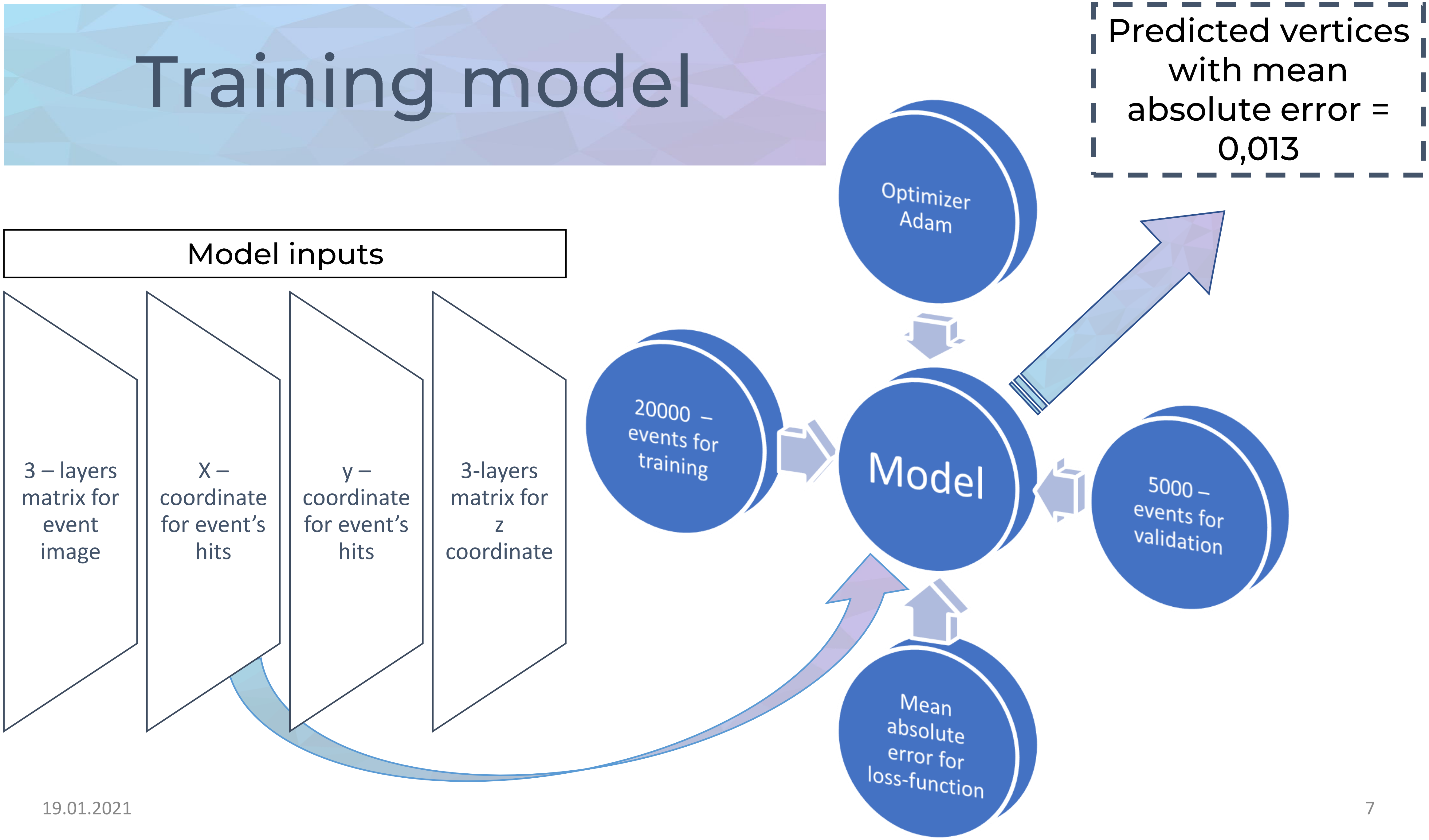


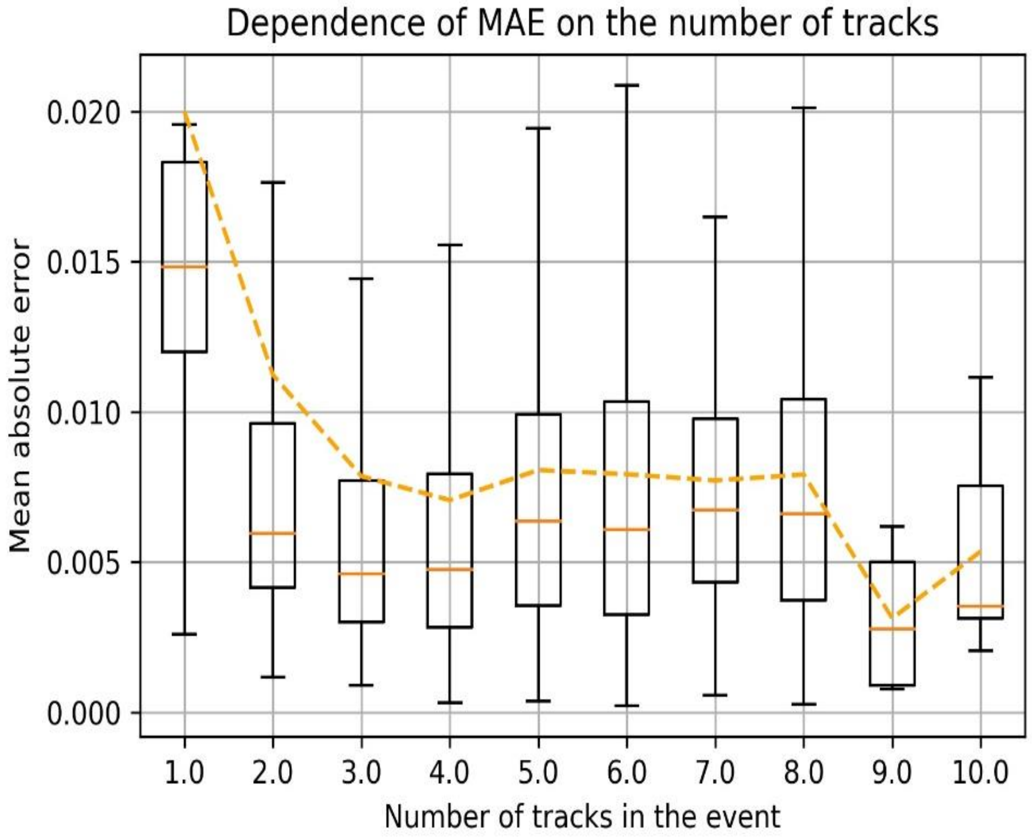

Events to binary matrix

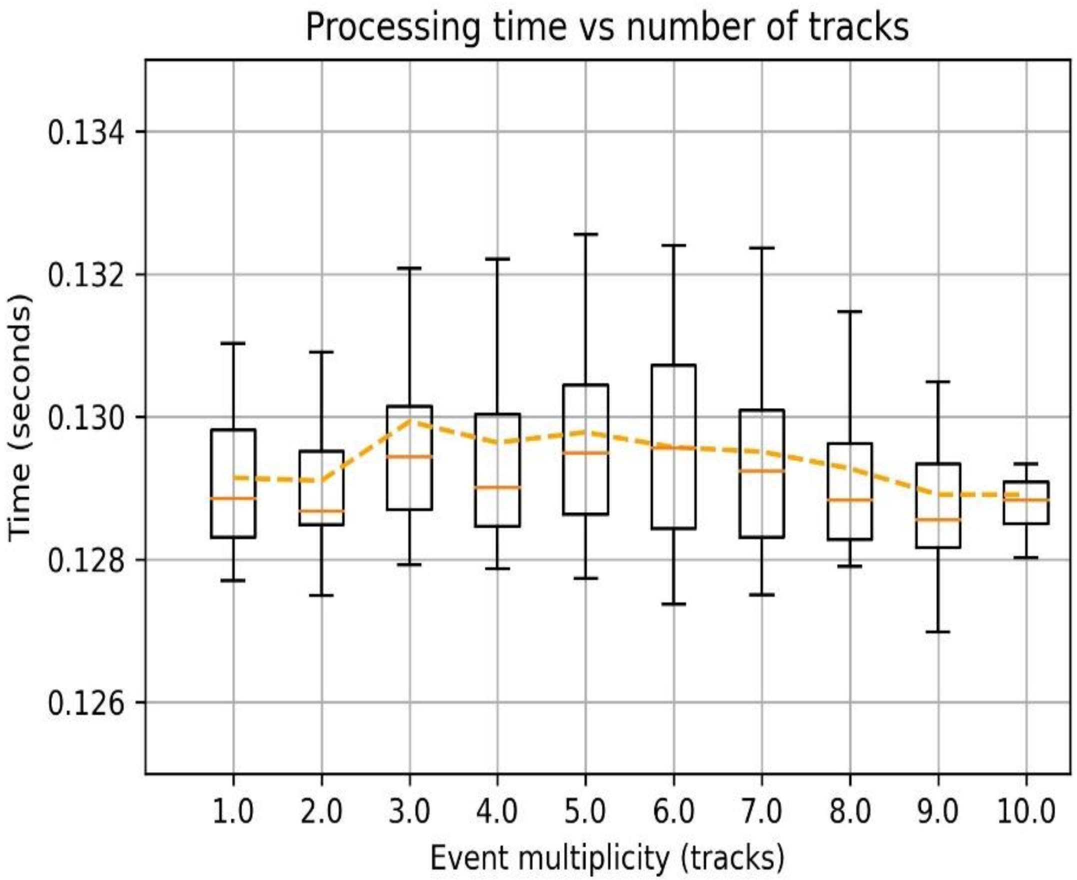



Add features for coordinates

# Training model






Inference speed on a single Nvidia Tesla T4 equals to 80 events/sec

# Improving the model